Survey on RPL enhancements: a focus on topology, security and mobility

Patrick Olivier Kamgueu[a,b,*], Emmanuel Nataf[a], Thomas Djotio Ndie[b]

[a] *University of Lorraine, UMR 7503, Nancy – France*
INRIA Grand Est – MADYNES Team
[b] *University of Yaounde 1, Cameroon*
LIRIMA, IoT4D Team

Abstract

A few years ago, the IPv6 Routing Protocol for Low-power and Lossy Networks (RPL) was proposed by IETF as the routing standard designed for classes of networks in which both nodes and their interconnects are constrained. Since then, great attention has been paid by the scientific and industrial communities for the protocol evaluation and improvement. Indeed, depending on applications scenarios, constraints related to the target environments or other requirements, many adaptations and improvements can be made. So, since the initial release of the standard, several implementations were proposed, some targeting specific optimization goals whereas others would optimize several criteria while building the routing topology. They include, but are not limited to, extending the network lifetime, maximizing throughput at the sink node, avoiding the less secured nodes, considering nodes or sink mobility. Sometimes, to consider the Quality of Service (QoS), it is necessary to consider several of those criteria at the same time. This paper reviews recent works on RPL and highlights major contributions to its improvement, especially those related to topology optimization, security and mobility. We aim to provide an insight into relevant efforts around the protocol, draw some lessons and give useful guidelines for future developments.

Keywords: RPL, Low-power and Lossy Networks, topology optimization, mobility, security, Routing Protocol, Internet of Things

1. Introduction

Low-power and Lossy Networks (LLNs) are classes of networks where nodes are largely resources constrained. They have limited processing power, work with a scarce memory and mainly operate on batteries or rely on an energy scavenging unit. Those nodes are interconnected by lossy links that support only low data rates and their state is usually unstable with low packet delivery rates. An LLN supports a wide range of application domains. Those include home (e.g. lighting, remote video surveillance, window shades, alarm systems, healthcare appliances) and building automation (e.g. HVAC [1] systems, fire, physical security devices, lift control) scenarios. Sensors and actuators are remotely monitored and controlled to provide a safe and comfortable environment [1]. The use of these networks with nodes placed outdoor in an urban environment is widespread. For instance, their application in smart cities allows them to measure and report data related to meteorology (temperature, humidity, pressure, UV index, wind direction and strength) or pollution, and manage urban devices such as street or traffic lights. Also, in smart grids they enable the remote monitoring of electric, gas and water smart meters through a city-wide distributed network [2]. This allows to the identification of peak loads and match energy production to household demands in smart grids systems [3]. In the industrial field, LLNs enable users to increase the amount of information collected and the number of control endpoints (fuses, pumps, luminaries, HVAC status) that can be remotely managed. As a result, they improve the productivity and the safety of plants while increasing the efficiency of the workers.

These networks are primarily part of the Internet of Things (IoT) paradigm [4]. In the latter, various physical entities are connected to the virtual world and receive their orders from the Internet [5]. Use cases related to urban, industrial, home and building automation applications mentioned above, have specific requirements formally expressed in Internet standard documents [6–9]. These requirements are different from those of traditional wired or wireless ad hoc networks. They concern, but are not limited to traffic flow characterization, scalability, latency, network dynamicity, manageability, stability, convergence time and support to mobility.

*Corresponding author
Email address: patrick-olivier.kamgueu@inria.fr (Patrick Olivier Kamgueu)

[1] Heating, Ventilation and Air Conditioning: technology of indoor environment comfort with the goal of providing thermal and acceptable indoor air quality.

Thanks to the ability to carry IPv6 packets on top of low-power and low-rate IEEE 802.15.4 physical layer through the 6LoWPAN protocol [10], the IETF [2] designed a routing protocol tailored for LLN, namely IPv6 Routing Protocol for Low-power and lossy networks (RPL). The latter was standardized in RFC6550 [11] and optimized for point-to-multipoint traffic flow. It also provides mechanisms for multipoint-to-point, as well as point-to-point communications [12, 13]. Packet processing and forwarding were separated from routing optimization goals through the *objective function* (`OF`). Several metrics are intended to be used with the protocol during the topology building step [14]. So far, only the classical hop count and a popular link reliability metric known as `ETX` [3] [15] are considered in the standardized `OF`s [16, 17]. The network designer is free to shape and implement new `OF`s. So, several works have investigated the use of other routing metrics (different from the aforementioned). They consider energy consumption of nodes [18], avoid network bottlenecks [19], favor high throughput paths [20, 21] or other optimization criteria. Several other issues have not been well addressed. For instance, the combination of several metrics to capture more than one network characteristic to meet Quality of Service (`QoS`) has received little attention [22–24]. Some other works have studied RPL parameters under mobility scenarios and propose enhancements to improve the protocol in that context [25–29].

The last but not the least challenge is related to security. Indeed, given the *lightweightness* of RPL, the constrained hardware involved and the open nature of wireless medium, ensuring data privacy and securing communications among nodes are challenging issues. It is then necessary to identify main security threats and RPL inherent vulnerabilities and enable countermeasures to mitigate them.

This paper surveys main contributions proposed by the research community to improve RPL, the de facto standard for routing in the IoT. We focus on works optimizing the network survivability while considering the application goal, security of communications and mobility of nodes. We also review some open research issues and identify lessons learned from recent works in this field. The paper is organized as follows. Firstly, we recall some RPL fundamentals and review proposed `OF` implementations that optimize the routing topology building. Then, we highlight main security threats and efforts to mitigate them in the RPL context. Section 4 investigates mobility considerations, followed in section 5 by lessons learned from the covered topics, as well as discussions about issues that remain open. Finally we conclude the paper.

[2] Internet Engineering Task Force

[3] Expected Transmission count: It's the expected number of transmissions needed to successfully transmit a packet on a link.

2. Topology optimization

RPL was proposed as the routing standard for LLNs a few years ago [11]. It uses many mechanisms and techniques optimized for constrained devices. Below, we highlight relevant aspects of this protocol and discuss optimization goals when building the routing topology.

2.1. RPL relevant aspects

Designed as a distance vector routing, RPL topology is organized as one or more Destination Oriented Direct Acyclic Graph (`DODAG`), each rooted at a single point: the `DODAG` root (also known as the sink in WSN vocabulary). Typically, this special node acts as an IPv6 Border Router and connects LLN with the outside world (Internet or any other networks) from which it can receive orders or to where the collected data will be managed. Nodes may operate through one or more RPL instances consisting each in an optimization goal that relies on the application objective, later translated as the `OF`.

Figure 1 illustrates a simple RPL topology that consists of a single `DODAG`. Initially, only the root is part of the RPL active topology (Fig. 1a). It periodically broadcasts configuration parameters in its neighborhood through a new dedicated ICMPv6 control message, the `DODAG` Information Object (`DIO`). This message conveys necessary parameters for the topology building and maintenance. The most relevant are the `DODAG ID`, instance `ID`, version number, node's rank, mode of operation (`MOP`), timer parameters, `OF` code point (`OCP`) and metric values. As soon as a node receives several `DIO`s originating from different sources (Node 4 in Fig. 1b), it selects one of them as its preferred parent that also acts as the next-hop to reach the sink (upward route). The `OF` discussed below in §2.4 governs how nodes select their parents according to metrics transformed into a rank value. The rate at which `DIO`s are spread onto network is regulated by a timer [30] based on the *Trickle* algorithm [31]. The aim is to speed up the dissemination of correct/up-to-date information when inconsistencies occur, but slow their propagation down in a steady stage. As consequence, *Trickle* typically allows fast recovery while ensuring low overhead. Inconsistencies could be the divergence of configurations information, nodes mobility (significant changes in neighborhood), loop detection or any other unusual situation.

RPL also defines some other ICMPv6 control messages. Among others, the Destination Advertisement Object (`DAO`) allows a node to advertise its prefix for downward routes establishment (Fig. 1c). It must be acknowledged by the parent through a `DAO-Ack`. Regarding the `DODAG` Solicitation Object (`DIS`), it's used when a node actively wants to claim fresh configuration parameters without waiting for the *Trickle timer* to operate.

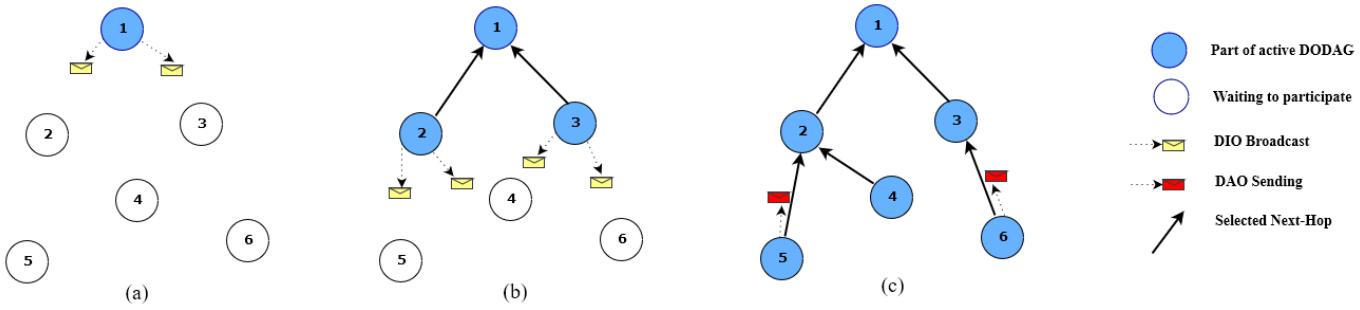

Figure 1: DODAG Construction. (a) RPL start-up: root node broadcast configuration parameters through DIO. (b) Surrounding nodes start participating in the DODAG under construction, select root as their preferred parent and spread their own configuration parameters. (c) When all nodes are engaged, the DODAG is built. Depending on mode of operation, nodes can advertise their prefix (through DAO) and build downward routes.

2.2. How to deal with traffic patterns?

Depending on how the application would like to access (or not) the internal nodes, RPL allows several modes of operation encoded into the DIO signaling messages using a three-bit field. When the upstream traffic is the single flow pattern, the MOP flag is set to 0 which means that no downward routes is maintained by the RPL internal routers nor by the sink. If the application wants to build reverse paths to reach the internal nodes (also referred as *routers* when they are not leaves), the sending of DAOs is enabled to install routes toward these nodes. How the DAO messages are handled by nodes (the root or routers) allows to distinguish two ways of keeping routes on them. In a *non-storing* mode (MOP flag set to 1), only the root keeps the downward routes. Later, data packets are "source-routed" from root to reach any RPL internal node. The *storing* mode allows each node to store the downward route information for all its descendants (*i.e.* located in its sub-DODAG). Depending on whether the multicast is supported or not, the MOP flag for the *storing* mode is set to 3 or 2 respectively.

2.2.1. Extension to cope with P2P flows

No matter which MOP is used, except for the *no downward* mode, when two internal nodes want to communicate, the packet travels upwards along the DODAG until it reaches a common ancestor that maintains a route to the destination, and then the packet goes down. As a result, this *up* and *down* routing along the topology causes the point-to-point (P2P) routes between arbitrary pair of nodes to be suboptimal. Moreover, these routes may lead to traffic congestion near the DODAG root. There are many application scenarios that rely on P2P communications for their operations. For instance, some use cases in home and building automation involve a device (*e.g.* remote control or motion sensor) to communicate with devices (*e.g.* lamp modules) to which it does not already have a route. Things are exacerbated when energy and latency are constraints to be meet. They are quite difficult to satisfied by P2P routes along the existing DODAG as they involve traversing many routers than necessary to reach the destination.

Xie *et al.* [32] provide intensive simulations for a sample of 1001 node topology to evaluate the gap between the shortest cost routes available in the network and the *up/down* routes along the DODAG through a common ancestor. This inadequacy of DODAG-based P2P routing, particularly appalling when the source-destination pairs are relatively close leads to the development of a new mechanism that helps RPL to deal with P2P flows.

That said, an extension of RPL has been proposed [12, 13] as an experimental standard. The latter is based on a reactive scheme that can provide shorter P2P paths on-demand without necessarily going through the sink. So, when a sensor S needs to discover a shorter P2P path to another sensor D, it initiates a route discovery process through information piggybacked on DIOs and disseminated throughout the network using a temporary DODAG rooted at S. A new RPL mode of operation referred as P2P *route discovery* mode is specified with the flag set to 4. Once the route is discovered and established, the temporary DODAG ceases to exist, and packets between S and D can flow through the installed route using source or hop-by-hop routing scheme depending on the configuration made. Experiments on a real-world deployment [33] corroborate the findings obtained from previous works using simulations and theoretical aspects [32]. P2P traffic flows are now using paths that are much shorter than those that would have been selected through the legacy RPL.

The main drawback of the proposed extension is the extra burden borne by the nodes and traffic overhead due to the maintenance of an additional (albeit temporary) DODAG. However, this overhead is acceptable for application scenarios where P2P flows are prevalent.

2.2.2. Interoperability between RPL modes of operation

When maintaining downward routes in a RPL topology, both the *storing* and the *non storing* MOPs have their own strengths and weaknesses. While the *storing* mode requires each node to store route information to all desti-

nations in its own sub-DODAG, leading to scalability issues and memory limitations, the *non storing* mode introduces communication overhead near the sink and causes source routing headers to increase the packet size depending on the network depth. It becomes obvious to investigate what would happen when both modes coexist in the same RPL network in order to leverage their benefits while avoiding drawbacks. Ko *et al.* [34] have shown that there exists a serious connectivity problem resulting in network partitions when these two MOPs are mixed within a single network using a standard-compliant implementation of RPL. To address the aforementioned problems while preserving a high bidirectional data delivery performance, they propose DualMOP-RPL, an enhanced version of RPL, which supports nodes with different MOPs for downward routing. The new RPL version releases some restrictions and introduces several modifications both in the DAO message format and in the control/data planes that enable heterogeneous MOP environments to interoperate gracefully in a single RPL network. Although the proposal achieves packet delivery identical to the homogeneous MOP scenarios both through simulations and real testbed environments, the network size is only limited to 25 nodes, moreover, the introduced modifications are far from the standard track.

To reduce the memory usage related to the *storing* mode of operation when the network size increases, Gan *et al.* design MERPL [35], a scheme which leverages the RPL *non storing* mode while reducing communication overhead compared with the pure *non storing* mode. The proposed scheme ensures that the number of routing table entry stores in a node does not exceed a predefined value N. When the value is about to be exceeded, MERPL opportunistically uses *non storing* features to overcome the memory limitation. More specifically, as soon as a node suffers lack of memory, it removes some entries in its routing table and then it informs its child that is used as next-hop for these entries. Later, the source routing is used from the DODAG root to reach destinations related to deleted entries. A node with exactly N entries (called a *storing node* in this scheme) sends its complete routing table to the DODAG root to enable this *reachability* through the child node. The main advantage of this proposal is the ability to vary the size for storing routing entries from $N = 0$ (pure *non storing* mode) to a sufficiently high value of N (*storing* mode) depending on nodes' memory resources. Unfortunately, MERPL lacks experiments on real testbed to evaluate the protocol behaviors on real channel conditions and constrained resource environment.

For the same purpose of dealing with memory limitations, instead of mixing these two MOPs in the same network, Kiraly *et al.* propose D-RPL that takes advantage of multicast communication scheme to bypass path-agnostic areas of the DODAG [36]. When a node cannot store (nor propagate it upward to the root) a received DAO coming from its sub-DAG, it sends back a DAO-NACK to the originator. The latter that failed to advertise the DAO joins a special multicast group, whose address is used by the root to send data for destinations for which it does not have a route. The normal RPL unicast operation is resumed as soon as the packet reaches a group member knowing a route to the destination. Compared to the previously mentioned solutions, D-RPL requires a minimal disruption on the legacy protocol, unfortunately the authors do not provide any experimental results which are crucial for evaluating the performance (especially the energy cost) of the multicast used.

2.3. Improving the link quality estimation

It is clearly stated in the standard that: "*RPL expects an external mechanism to be triggered during the parent selection phase in order to verify link properties and neighbor reachability*" [11]. The rationale behind this decoupling of link quality estimation (LQE) techniques from RPL mechanisms for topology construction and path calculation is the desire for the standard to ensure a high degree of implementation flexibility while reducing the implementation complexity. However, several works show through an in-depth analysis that this design tradeoff leads to routing inefficiencies which affect RPL performance [3, 37, 38]. Indeed, because RPL has no control on the LQE process, the routing decision could be based on outdated or inconsistent link statistics [38]. For instance, Ancillotti *et al.* show that, under certain scenarios and depending on the implementation, RPL experiences high packets loss because some nodes maintain unreliable routes even though reliable alternative routes exist [3, 38]. The main reason for this is that when a node selects a parent with a bad link, it may be unable to switch to a better parent because the implementation used adopts a conservative approach for link estimation (only the links that are currently being used are evaluated).

To address this RPL's lack of accurate knowledge on the link quality, authors in [38] adopt cross-layering design approach to provide some routing optimizations through enhanced link estimation capabilities and efficient policies for neighbor table management. They propose RPLca+ that use a hybrid approach for link estimation. The latter combines both active probing of new discovered links and passive link monitoring that opportunistically exploits existing data traffic to measure link qualities. Furthermore, the routing engine dynamically activates the most efficient link estimation technique depending on node status and the characteristics of the link to be monitored. Performance evaluations conducted through simulations and real-world testbed show that RPLca+ noticeably outperforms Contiki's implementation of RPL (ContikiRPL [39]) on packet delivery rates.

To be lighter, Trickle-L^2 rather leverages the *Trickle* mechanism used for topology maintenance to simultaneously estimate link qualities while disseminating routing

information [40]. More specifically, instead of using a passive monitoring strategy, the link estimation procedure exploits routing signaling messages for conveying probe information on links quality. These probes are piggybacked into DIOs thanks to a sequence number which updates link quality information of all neighbor nodes, and they are regulated through the *Trickle* algorithm. The main goal is to incur a small overhead compared to classical active link monitoring approaches that use unicast frames.

More recently, Kim *et al.* [41] noted that the most representative RPL implementations, such as ContikiRPL [39] and TinyRPL [42, 43], update link qualities of parent candidates only through transmission results of upward traffic. While it is true that RPL was designed to mainly support upward traffic, making reliability and efficiency of routes for downward traffic-centric applications dependent on not only uplink quality but also on frequency at which upward packets are transmitted, there are several new IoT applications which exhibit diverse traffic patterns [44–46]. Authors show that ContikiRPL experiences a significant performance degradation when delivering downward-centric traffic [41]. So, they provide DT-RPL an improve RPL implementation in Contiki which updates link quality through both upward and downward traffic to support Diverse Traffic patterns (hence the DT). Extensive evaluations through both simulations and indoor testbed reveal that DT-RPL significantly outperforms ContikiRPL, in terms of packet delivery ratio and control overhead and achieves reliable delivery of bidirectional traffic regardless of traffic patterns.

In order to reduce the overhead and energy incurred to accurately/efficiently estimate the link quality, some authors take advantage of machine learning strategies when driving the monitoring procedures. So, a recent link quality monitoring scheme for RPL called RPL-Probe [47] has been designed for leveraging reinforcement learning techniques based on multi-armed bandit model in order to maintain up-to-date information about link quality while promptly reacting to sudden and unpredictable topology changes (*e.g.* mobility). On the other hand, authors in [48] propose the fuzzy logic for link quality prediction and use fuzzy rules to combine multiple link metrics while compensating for the uncertainties in the wireless channel conditions. There are many other link quality estimators provided in the literature that cope with unique characteristics of WSNs, an interested reader is referred to [49] for a comprehensive survey.

To sum up, RPL as a layer 3 protocol decouples the LQE mechanism from its general scope, indicating that such a mechanism should preferably be reactive to data traffic in order to minimize the overhead. This design choice has been adopted by the most widespread RPL implementations, but investigations show that the protocol in such conditions fails to achieve high delivery rates. As a result, when decoupling the neighbor management and nodes' reachability from the standard goal, RPL must relies on a powerful and well-designed LQE and neighbor management mechanisms to provide good performances [50]. This makes the neighbor table management not just an external mechanism to RPL, but an essential component that significantly effects the protocol performances.

2.4. On the design of enhanced OFs

An important point when building the routing tree is how metrics values and constraints conveyed by DIOs are used to compute ranks. According to the gradient-based approach, rank value (node distance relative to sink) should monotonically increase when moving from DODAG root to leaves to guarantee a loop-free topology. These optimizations are conducted through the OF. So far, the standard scope lies on two OFs:

- OF0 implements a simple routing scheme based on the hop count [16].
- MRHOF selects routes that minimize the ETX (additive metric) while using an hysteresis to reduce transient churns [17].

A set of metrics suitable for LLNs is specified [14], divided into link (ETX, latency, throughput) and node (hop count, energy) metrics. Defining news or use the previous in novel OFs is left to the network designer. Below we investigate recent works in this direction.

2.4.1. Single metric OF

To address congestion problems that occur in case of heavy data traffic, Al-Kashoash *et al.* design a congestion-aware (CA) RPL that considers buffer occupancy as the route selection criterion [21]. Albeit ETX informs about how the link or channel is congested, it does not encompass node congestion. Indeed, many packets are discarded when the buffer overflows. So, CA-OF improves the delivery ratio by avoiding paths with congested nodes in heavy traffic scenarios. However, an opposite effect of this proposal is the routing instability that can result.

In a similar way, Di Marco *et al.* take advantage of cross-layer design and propose a Medium Access Control (MAC)-aware routing metric that takes into account complex interactions between MAC and routing [20]. Two metrics labelled R and Q that extend ETX by considering effects of contention and MAC parameters were provided. The R-metric includes packet loss due to the link layer contention and intends to improve the delivery ratio. As for the Q-metric, it determines the amount of traffic that must be handled by a given node, aiming to balance the traffic distribution and therefore the network lifetime. Authors compared their schemes against the back-pressure algorithm [51] and demonstrated that they perform better both in end-to-end reliability and power consumption.

Kamgueu *et al.* instead try to maximize the network lifetime and consider the node remaining energy as the only metric used to build routing paths [18]. Knowing their initial energy capacity, nodes add up time periods spent in each state or activity (transmit, receive, idle, sense, compute). Later, the node uses the current drawn in corresponding states (according to the technical data sheet of the device) and derives the remaining energy capacity [52]. The path cost is designed as the energy value of the weakest node on the path. So, to avoid further hurting the weakest node, energy-`OF` selects as the next-hop the neighbor for which the path to the root has the higher cost (*i.e.* the largest minimum lifetime) among all admissible paths. The main drawback of this scheme is that the transmission reliability is completely left out.

Still considering the maximization of network lifetime, Iova *et al.* define a new metric, the Expected LifeTime (`ELT`) [19]. The latter includes in its computations, both the amount of traffic and the link reliability when estimating how long a node could live before exhausting its battery. Rather than trying to avoid the weakest nodes (also called *energy bottleneck*), the routing scheme seeks to unevenly spread traffic over all *bottlenecks*, according to their respective strengths. The underlying idea is to balance the lifetime of the set of *bottlenecks* for a given node. Although promising great performances, the proposed solution (as well as the associated heuristic) induces a high computational overhead that is unfeasible in LLNs environment.

To deal with issues related to time critical applications, a metric that minimizes end-to-end delay is proposed in [53]. Bearing in mind that the link layer in an LLN relies on duty cycling mechanisms that keep nodes in sleep state more often for energy efficiency purposes, the impact on the overall latency may be significant. Trying to achieve low latencies while running low duty cycle on nodes could be conflicting. Authors introduced some enhancements (support different sleeping periods) at the link layer level and designed an `OF` that reduces delay even for nodes far away from the root.

2.4.2. Composite metric `OF`

Routing schemes reviewed in the previous section optimize the network topology according to a single routing metric. However, some applications need to achieve several goals to meet `QoS`, or combine some requirements (reliability, delay, etc.) while considering LLN limitations (*e.g.* energy). It then becomes sometimes necessary to consider several criteria when designing an RPL `OF`. The problem of combining metrics with RPL is left out of the standardization process, but some recent works have addressed the issue. Karkazis *et al.* provide a systematic analysis of properties that the basic metrics must hold to be combined in an additive or lexicographic manner [23]. Following the routing algebra established through Sobrinho and Yang works [54, 55], to ensure that routing protocol hold the features of convergence, optimality and absence of loop, metric resulting from a combination must be *isotonic* and *monotonic*.

Thanks to the previous algebra, Velivasaki *et al.* designed a composite routing metric that reflects both link reliability and trustworthiness of the next hop [24]. The former relies on `ETX` and the second on a new trust-aware routing metric named `PFI` (Packets Forwarding Indication). The `PFI` metric aims to cope with nodes acting either selfishly (save their energy) or maliciously (hijack traffic). To implement this metric, after sending a packet to a neighbor, the node enters in a promiscuous mode and waits for a given time to listen whether the selected neighbor has forwarded its packet or not. The node's trust knowledge is built according to the estimated probability of each neighbor to forward its packets. Given that isotonicity and monotonicity properties hold for both metrics (`ETX` and `PFI`), authors evaluated performances obtained through various weights (resp. orders) of additive (resp. lexicographic) composition approach. They found that composite metrics always behave better than single metrics. However, an additive composition outweighs the lexical approach, when both link and trust reliability are pursued.

The aforementioned composition strategies, either constrain types of basic metrics that can be combined (additive approach) or exhibit the properties of the first metric of the composition most of the time (lexicographic approach). Kamgueu *et al.* propose to leverage fuzzy inference-based reasoning to combine several metrics even when they exhibit antagonistic properties [22]. Moreover, this approach requires a small memory footprint that is well-suited for LLN nodes. It also allows a fine-grained parametrization of basic metrics according to the strength the designer wants. Authors used the proposed strategy to implement an RPL `OF` that combines energy, `ETX` and delay. They also provide a real deployment scenario of their proposition.

2.4.3. Summary on new routing metrics used to design `OF`

Besides the two objective functions provided by the standard track to govern the construction of the DODAG, several others, more and more elaborate were proposed. They may seek to optimize a given criterion or intended to be combined in a non-trivial way to find a good tradeoff between several criteria. This avoids jeopardizing network resources while achieving the application goal. Table 1 summarizes these main contributions.

3. Securing the protocol

Protocols in LLNs are exposed to several threats. Besides the fact that they are based on IP open stack, they used a wireless medium which exposes them to passive eavesdropping. Furthermore, several attacks may originate from inside the LLN, therefore all the traditional

Type	OF name	ref.	Metrics used	Captured effect	Measurement tech.	Target performance
Single metric	CA–OF	[21]	Buffer occupancy	Buffer free spaces on traversed nodes	Passive monitoring	Reduce lost packets rate due to buffer overflow.
	[R & Q]–OF	[20]	MAC reliability (R) Traffic load (Q)	Back-off & re-transmis^o Amount of traffic	Both passive (local informat^o only)	Achieve high throughput Provide load balancing
	Energy–OF	[18]	Residual energy	Online estimat^o of node remain. energy	Passive (local info.)	Delay the battery depletion of the weakest nodes
	ELT–OF	[19]	Expected lifetime	*Bottleneck* ELT by considering several params	Active probing of certain parameters	Maximize lifetime of the most constrained nodes
	delay–OF	[53]	Average delay	Time receiving packet & delivery to next-hop	Passive+piggybacking on control msg.	Achieve low latency even for nodes away from root
Composite metric	Additive & Lexicographic	[23, 24]	ETX combined w/ PFI according to Add. or Lex.	Link reliability & neighbors trustworthiness	Active probing	Cope with selfish nodes while avoiding unreliable links.
	OF–Fuzzy	[22]	Combination of energy, ETX and delay metrics	Weighted QoS of targeted metrics	Active or passive depending on selected metrics	Find a tradeoff between reliability, latency and network survivability.

Table 1: **Summary of proposed RPL metrics for topology optimization**

communication encryption techniques are no longer efficient to ensure data security. It is then necessary to identify main threats and implement countermeasures to ensure that the routing protocol operates properly. These measures must be implemented in such a way that they comply with security standards as the ISO 7498-2 security reference model [56] which includes C.I.A.A (Confidentiality, Integrity, Authentication and Availability). Due to the particularity of LLNs, implementation of such security measures meets specific issues. In addition to their constrained hardware or limited physical security, protocols and deployed services are intended to scale to many nodes. Moreover, the autonomous properties of operations (self-organization and self-configuration) make the key management and their deployment more complicated.

3.1. RPL basic security and self-healing mechanisms

In its standard version, RPL provides a few security measures and self-healing mechanisms to ensure proper network operations [11]. The security mechanism natively provided consists of data confidentiality and integrity. Indeed, an RPL network admits three possible security modes: unsecured, pre-installed and authenticated. While the unsecured mode relies on link/transport layer security materials (if any) to secure exchanges, the last two modes use pre-shared keys. Contrary to the pre-installed modes where nodes fully join the network thanks to pre-shared keys, in authenticated mode, they join the network only as leaves until they get a second key from an authentication authority before acting as routers.

RPL self-healing mechanisms ensure that the protocol operates safely and can overcome certain inconsistencies by itself. One of these mechanisms is loops detection and repair while carrying data. The former is done through a data-path verification and validation tool. Indeed, every data packet crossing an RPL-based network includes an IPv6 extension header (RPL hop-by-hop option) carrying flags on how the packet must be handled by RPL routers [57]. Later, these flags are processed by nodes to determine whether inconsistencies have been detected and what are the appropriate measures to take (local or global repair as instances). This improves RPL resilience and helps to counter some of the attacks discussed below.

3.2. Main security threats and their mitigation

As stated earlier, implementation complexity and memory size are a core concern in LLN environments. However, many efforts have been done to adapt sophisticated security methods based on encryption and key materials [58]. For instance, recent implementations aim to securely connect constrained nodes in an LLN with the Internet using a lightweight compressed IPsec [59], a lightweight DTLS [4] [60, 61], or IEEE 802.15.4 link-layer security [62]. But much more remains to be done, including the standardization process. Furthermore, the RPL standard clearly mentions that the authenticated security mode must not be supported by the symmetric encryption, although the asymmetric cryptography is not currently supported by RPL (therefore authenticated mode also) [11, 58].

[4] Datagram TLS (Transport Layer Security): aims to provide the same security services as SSL (Secure Socket Layer) under UDP and prevent eavesdropping, tampering, or message forgery.

Even when network is protected using inherent secure modes or use link/other layers security mechanisms [62], the RPL topology is not free from attacks that originate from internal nodes. Moreover, an intruder can eavesdrop the network or exploit known vulnerabilities, then gain access to a shared key or bypass the traditional encryption-based security protections. So, we only consider attacks where the intruder is part of the active RPL topology, also referred as byzantine attacks. Many works investigated such attacks [63–66] and others propose some countermeasures to mitigate them [67–70]. Moreover, authors in [71] provide a systematic classification of main threats related to RPL-based networks. Depending on the intruder's aims, attacks can be of several types. Those targeting network performances or exhausting its resources. Also, they can aim to act on network topology and hijack the traffic. Below we look into the most relevant of them.

3.2.1. DODAG inconsistency attacks

A malicious node can take advantage of the self-healing mechanism introduced above. Indeed, RPL fields carried in IPv6 hop-by-hop options header include both the sender rank and some flags. A 1-*bit* field named `'O'` flag indicated the direction in which the packet is flowing: down (flag set) or up (clear). A loop occurs, if the sender and receiver rank relationship does not match `'O'` flag value. A node detecting the mismatch must set the packet's `'R'` flag (Rank-Error) and could drop it. One inconsistency for the same packet along a path is not critical, however, if inconsistency is found while the `'R'` flag was already set, the packet should be dropped, and the *Trickle* timer reset.

Intruders could manipulate the RPL extension header and would force nodes to reset their *Trickle* timer more often. Indeed, by setting both `'O'` and `'R'` flags inappropriately, it would cause packets to be dropped at the next hop. Local and global repairs mechanisms involved will lead to a Denial of Service (DoS) due to the high number of control messages. As a result, local instabilities and excessive power consumption occur, caused by an increase in the DODAG maintenance overhead and the reduction of channel availability.

To counter this behavior, IETF [57] advocated limiting the rate of *Trickle* timer reset, caused by the reception of RPL mismatch messages to a fixed threshold (20 per hour by default). Once the threshold is reached, all subsequent packets with erroneous headers are dropped during a time interval, but the *Trickle* timer is not reset. Mayzaud *et al.* [70] discussed this arbitrary setting of the threshold and propose to tune it dynamically according to the attack pattern (*i.e.* the level of aggressiveness) and network conditions. The latter is designed so that it drops quickly when the attacker is aggressive and increases slowly once attacks stop. With this scheme, authors show that they drastically improve the control packets overhead (thus energy and network stability) compared to the RPL inherent mitigation strategy.

3.2.2. Black and Greyhole attacks

In a blackhole attack, like a hole sucking in everything that goes through, a malicious node silently drops all data packets that it's supposed to forward. If the intruder has a high-level position in the RPL topology hierarchy, a large part of the network could become isolated and then unreachable. Selective forwarding, also known as greyhole is a variant of this attack. The intruder just discards a part of traffic that goes through it. Consequences of both attacks on an RPL-based network were investigated in [72]. Authors found that some performance indicators (high number of `DIO`, signalization rate, packet loss and delay) point out the presence of such attacks.

RPL does not provide means to prevent such attacks, but Raza *et al.* propose `SVELTE`, an Intrusion Detection System [5] (`IDS`) that monitors the network and detects inconsistencies related to them [67]. Thereafter, `SVELTE` can identify malicious nodes and instructs their neighbors to blacklist them, and therefore, this prevents intruders from participating in the active RPL topology.

3.2.3. Sinkhole attack

Here, the malicious node alters control packets and advertises a good position in the DODAG (rather than the one that reflects actual network conditions). The intruder's purpose is to attract all or a significant portion of traffic. This attack is very damaging when combined with the previous attacks (Black or Greyhole). Indeed, because the intruder has gained a good position in the network, more traffic can be discarded. In RPL-based network, the easiest way to launch such an attack is to advertise a low rank value (*fake rank*), regardless of the intruder node's position in the network. As a result, surrounding nodes will select the attacker as the preferred parent, although it does not fulfill the application goal.

Several defense strategies can be implemented to counter this attack. `VeRA` [73] was designed as a security scheme that prevents an internal node to advertise a *fake* rank. Nodes send signed `DIO` according to their rank, where the message signature is built through a one-way hash chain. The receiving node can authenticate control messages and determine if one node along the path has advertised a *fake* rank. `VeRA` was improved later by `TRAIL` [74] to consider rank replay and spoofing attacks. Both schemes guarantee that a node cannot falsify its position in the network hierarchy by more than one level. Weekly and Pister demonstrate that, combining `VeRA` with *parent fail-over* achieves high delivery ratio [75]. In their works, if the `DODAG` root does not receive enough data from a node according to a given threshold, it piggybacks node's identity in its future `DIO`. Nodes seeing their own identities in incoming `DIO`s blacklist their parents and look for another one. It's worth

[5] An IDS is a tool or mechanism used for detecting attacks against a system or network by analyzing activities in the system/network itself.

noting that IDS proposed in [67] also provides a solution for sinkhole detection. Unfortunately, it does not prevent attack without additional mechanisms.

3.2.4. Identity-based attacks

From eavesdropped traffic, a malicious node can get legitimate nodes identity. Clone ID attack consists in an intruder forging the identity of legitimate nodes onto another physical node. Later, the attacker uses these identities to replay legitimate nodes role. As a result, it can gain access to a larger part of the network or divert a voting scheme. A variant of this attack is the well-known Sybil attack that has been thoroughly investigated in IoT [76]. In this one, the intruder uses several logical entities on the same physical node, enabling to control over large parts of the network without deploying enough physical malicious nodes. These attacks are part of more general spoofing class attacks and they are usually used as premises to perform other attacks.

It's possible to mitigate these identity-based attacks by keeping track the number of instances of each identity or by coupling nodes identities with their geographic location. Indeed, no identity should be in several places at the same time. Clone ID attack in RPL network was studied in [64] and authors noted that RPL cannot counter related issues by itself.

3.2.5. Wormhole attack

A wormhole is a dedicated communication channel between a pair of nodes. This out-of-band connection could be wired or wireless links and it is used to tunnel data from one network location to another faster than through normal path. An attack based on this scheme requires two intruders that create the tunnel and exchange vital information through it. For instance, one of the attackers can replay all data coming from one network region to another far away and then disrupt normal operations or distort routing paths.

The wormhole attack is very hard to detect when switched on and off intermittently. Moreover, this attack is often launched in conjunction with another as the spoofing or sinkhole attack. Binding geographic information to neighbor tables and diversifying link layer key materials on separate network segments contribute to strengthen the security [64]. Another way to counter them is the use of a Merkle tree-based authentication scheme [77].

3.2.6. Version number attack

The version number is one of the most important RPL parameters embedded in RPL control messages. The DODAG root is the unique node that should increase the value to guarantee that topology does not become stale, but also to deal with some situations such as resolving inconsistencies associated with a routing loop (global repair mechanism). Hence, all nodes in a stable RPL topology must have the same version number. As soon as this number diverges somewhere, topology becomes inconsistent and nodes associate themselves with configuration parameters related to the highest-value of version number in their vicinity. Thereafter, they quickly propagate this information by resetting their *Trickle* timer. An intruder can exploit this behavior as a vulnerability and frequently increase the actual version number value associated with the topology so that nodes are forced to reset their timers very often due to the related inconsistency. Consequently, the resulting propagation of configuration parameters will consume too much energy and will create network instabilities as well as data loss.

Authors in [78] and [79] thoroughly studied impacts of this attack both on regular grids and random topology (that holds static/mobile nodes). They show that the severity of the attack is correlated with the position of the intruder with respect to the DODAG root. The further away the attacker is from the root, the more damaging the attack. The underlying reason is that, when the malicious node is far away, this delays the opportunity for the root to discover and resolve the version number inconsistency. The hash chain mechanisms previously used to counter rank-based attacks (Sinkhole) [73], [74] are also valid to prevent version number attacks. Indeed, these schemes use both rank and version number to generate the digital signature used for control messages authentication. Mayzaud *et al.* argue that cryptographic credentials induce an additional overhead that LLN nodes cannot afford, since they are already heavily resources constrained. Instead, they propose a distributed monitoring architecture that helps to detect this attack and identify malicious nodes involved [80]. In the proposed architecture, the network of monitored nodes (*i.e.* regular RPL nodes) is separated from that of monitoring nodes, which are more robust to perform monitoring tasks.

Table 2 summarizes RPL main security threats and provides relevant propositions to counter them, moreover the benefits of proposed solutions and their related costs are highlighted.

4. Mobility considerations

4.1. The RPL intrinsic mobility support

Initially, mobility was not a major concern in LLNs compared to other aspects such as the protocol *lightweightness* (resources constraint), scalability (network size) or security. Indeed, in RPL earlier requirements, devices should be mainly static, nevertheless some are intended to support a *reduced* mobility (*e.g.* wearable healthcare appliance, wheelchairs, vacuum cleaner robot) [8]. Likewise, in industrial field, it is expected that nodes located on vehicles (cranes, fork lifts) or moving parts (such as rotating components) reach speeds up to 35 km/h [7]. RPL must accommodate with the above requirements. For instance, to support mobility in a building automation system, it is

Attack name	Category	Exploited vulnerab.	Threat target	Countermeasure	Ref.	Overhead	Perf.
DODAG Inconsistency	DoS	Self-healing tools (ext. header, *Trickle* reset)	network lifetime & channel availability	Fixed threshold Dynamic threshold	[57] [70]	Very Low Low	Good Very G.
Black/Greyhole	Isolation	Trust exploitation	Network subset	SVELTE (*white/blasklist*)	[67]	High	Very G.
Sinkhole	Redirection	Trust exploitation	Attacker vicinity & Network subset	VeRA or TRAIL *Fail-over + rank auth* SVELTE	[74] [75] [67]	Low Low High	Good Very G. Very G.
Identity-based	Spoofing	Exposed node ID	Any node	Bind ID w/ GIS†	[64]	N/A‡	N/A
Wormhole	Replay	Access to network	Normal operations	GIS w/ neighbor tab, L2 keys per segment. Merkle Tree	[64] [77]	N/A	N/A
Version number	DoS	DODAG maintenance (global repair, up-to-date parameters)	energy exhaustion, network congestion & loss of data	VeRA or TRAIL Distributed monitoring architecture	[74] [80]	Low High	Good Good

† Geographic Information System.
‡ N/A = Not Applicable as no evaluated solution found.

Table 2: **Summary of attacks on RPL and their mitigation**

clearly advised in the standard documents that: "mobile nodes should not act as routers, while in motion (in order to minimize network dynamics). Rather, the mobile node should join the topology as a leaf. Furthermore, it must re-establish end-to-end communication with a static node within 5 secs after it ceases movement" [9].

Following a few years of RPL deployments and experiences, the expectations are met in a quite mobility environment, especially when communication takes place from mobile nodes upwards to the root. In contrast, the communication in the opposite direction (point-to-multipoint) is more difficult to maintain [82]. Indeed, as the mobile node move from one parent to another, downward routes established through DAO become stale much more quickly. As a result, downward communications are highly unreliable due to inaccurate route information.

Beyond initial expectations, RPL is nowadays considered as the de facto standard for the routing in the IoT. The trend shows that IoT application scenarios where devices are embedded in any kind of things in motion (across a denser topology of fixed nodes) will grow even more. A routing protocol that meets these new requirements should support more mobility even when nodes move at higher speeds than previously. Their application scenarios encompass assets tracking and wearable gadgets, robots and UAV (Unmanned Aerial Vehicles), cars, bus and trains equipped with more and more advanced sensors.

Although RPL has no restriction on the participation of mobile nodes upon routing topology, some recent works show that the routing protocol experiences some issues in a high mobility environment [25–27, 83–85]. This implies a high data loss rate for nodes involved in the motions. Indeed, when a mobile node moves out of the range of its current parent, the latter becomes unreachable and disconnected from the RPL topology. Later, the node could re-attach to the DODAG. This is done when it receives DIOs from a new neighbor having a better-quality path (in its new location) than its previous parent. Unfortunately, the DIOs emission rate is governed by the *Trickle* timer for which the next schedule may be triggered more than 2 hours later, according to the default RPL *Trickle* tuning [11]. On the other hand, the moving node could *aggressively* probe its vicinity, through DIS sending, to forthwith trigger DIOs from surrounding nodes. Once again, even when receiving new DIOs, the parent's switching is not guaranteed, because this one depends on ranks relationship with the previous parent (already out of range). To cope with these issues, improvements are required.

4.2. Mobility enhancements

Despite the fact that the rate at which reception of *fresher* DIOs by a node experiencing mobility should be increased, a mechanism that allows the node to quickly determine the previous parent unreachability is mandated. By default, RPL rely on external mechanisms for that. Neighbor unreachability detection incorporated in IPv6 neighbor discovery [86] or other MAC equivalent mechanisms can be used. Both are too complex and fail to quickly detect parent unreachability upon high mobility patterns. Furthermore, selecting a mobile node as the next-hop contrary to RPL best practices, would certainly lead to routing loops. Indeed, while moving the node can choose a prior node in its sub-DAG as a parent resulting

to a loop.

4.2.1. VANET: all field nodes in motion

Some works [83–85] focused on RPL tuning for VANETs (Vehicular Ad hoc NETworking). In that environment, all nodes (vehicles) are mobile within the network at different speed levels, except the sink played out by the Access Point. Since all nodes move and any could be selected as a router, it's necessary to have a mechanism that prevents loops formation. Lee *et al.* [83, 84] opt to piggyback parent's ID in DIOs. When a parent node loses network connectivity (moves away from its parent scope) later, it should discard all incoming DIOs stamped with its identity as parent's ID. GI-RPL [85], instead ranks nodes according to their geographical information (direction, distance, velocity) received from nodes by the protocol.

Due to the high network dynamicity and, in order to speed up the rate at which information about the RPL topology is exchanged to reflect frequent network reconfigurations or nodes re-association, Lee *et al.* disable the legacy RPL's *Trickle* in favor of a fixed (small but also periodic) timer value that governs the sending of DIO messages. On the contrary, GI-RPL uses an adaptive DIO period adjusted according to node velocity.

According to [83, 84], another change is to trigger ETX probing immediately when a node discovers new neighbors. So, it can determine whether to switch its parent to the new one, without delaying this process which would result in sub-optimal routes (more acute in high mobility scenario). Furthermore, when the RPL downward mode is enabled, as soon as the node selects its parent, contrary to the RFC recommendations (to delay), a DAO is immediately sent to notify the parent about node's routes as well as those of its sub-DAG. GI-RPL does not probe the neighborhood but it relies on dedicated infrastructure built with fixed sensors node on the roadside. Using a specific duty cycle scheduling (DCS) strategy, roadside nodes will sleep or awake depending on the direction of the moving vehicles.

All the aforementioned RPL parameters tuning improve packet delivery ratio and reduce disconnection time at the expense of additional control (DIO) messages overhead. The latter is correlated to the signaling messages transmission rate.

4.2.2. Mixed (static and mobile) nodes environment

When both static and mobile nodes coexist in the same network, introduce local operations (where mobile nodes are involved) rather than those that jeopardize the whole network operations would be much more efficient (energy expenditure, traffic overhead and network stability). As such, a way to distinguish static from mobile nodes is needed. Setting a new one-bit flag in DAO messages is sufficient for mobile nodes to advertise their status to parents.

To adapt the vicinity behavior of a mobile node to the latter's sustainability (related to surrounding nodes), Cobarzan *et al.* propose a *reverse Trickle* algorithm [27]. Contrary to the classical version, once it has connected a mobile node, the parent starts its timer interval with the maximum allowed value. The latter is halved every round as long as mobile nodes are part of the sub-DAG. According to the authors, the underlying idea is that a mobile node just connected to a new parent is likely to remain connected at the next period. So, it's unnecessary to schedule a DIO too early. Contrariwise, the more the mobile node spends time connected to the same parent, the more it's likely to move out of its coverage range. As soon as there are no mobile nodes in the vicinity, the parent switches from the *reverse* to the standard *Trickle*. Note that, a parent node can probe which mobile nodes are attached to it at any time, by advertising a DIO with an incremented value of DSTN [6] field. Thereafter, the parent counts the DAOs for which the *mobile* flag is set.

On the mobile node side, the reachability of the current parent is monitored. As soon as it's necessary, the mobile node invalidates its former parent (*i.e.* set the parent's rank to infinity locally) and claims for new DIOs by sending a DIS message. Rather than managing DIOs periods as above, Korbi *et al.* propose ME-RPL [25] that suggest leveraging the DIS message provided by RPL. The DIS interval period is dynamically adjusted depending on node preferred parent change rate (inconsistencies that mean mobility). The underlying principle is to leverage past knowledge to predict what happens in the near future. That is to say, a node that was inconsistent for several intervals has high probability to remain inconsistent in the future (owing to node or neighborhood mobility). The DIS interval is shrunk/enlarged according to the node's past behavior (number of preferred parent changes).

Gara *et al.* argue that the previous schemes do not consider random mobility scenarios (nodes pause times, velocity changes and random trajectories). In their proposed scheme mod-RPL [87], node's relative position with respect to its parent is detected by analyzing the difference between consecutive RSSI [7] values. The concept of *Time to Leave* (TL) was introduced, *i.e.* the time required for a mobile node to leave the radio range of its preferred parent. Depending on both nodes (moving node and its parent) velocity and direction, TL is estimated and timers that governed control messages issuing are adjusted.

Instead, Fotouhi *et al.* propose mRPL, where the mobile node continuously monitors the link quality (average RSSI) with its parent through various kinds of timers and

[6] Destination Trigger Sequence Number: Once received DIO with a higher value than its current recorded value of this field, node triggers sending a DAO message to its parent.

[7] Received Signal Strength Indicator: It's a measurement of how the power level that a RF device is receiving from the radio is, at a given location and time.

switches to a better parent candidate accordingly. They claim a hand-off delay less than a tenth of a second after loosing the next-hop connectivity [28]. Later, some improvements upon hand-off mechanism allow to distinguish the *hard* and *soft* hand-off [29]. Thus, mRPL+ is designed to support both, and the moving node can select a new link earlier (*i.e.* before disconnecting from the current one).

Unlike previous works that infer the node mobility from the node itself (status indication by bit flag in control messages) or link/neighborhood periodic monitoring, others [26, 88] manage mobility by assessing the mobile node's position or by using some anchors.

Gaddour *et al.* propose Co-RPL [26] a modified RPL version. To support mobility, Co-RPL divides the network into concentric circular regions around the DODAG root, called *corona*s. Each node (at a given time) belongs to only one *corona* identified by a unique identifier (*corona* ID). The preferred parent selection process is based on both *corona* ID and rank value. The *corona* mechanism aims to avoid routing loops by preventing a mobile node to choose as parent, nodes in the same *corona* as it. Also, no restriction is done on the ability of nodes (except the DODAG root) to move, but the DIO period is set statically before nodes deployment depending on the maximum velocity allowed. So, the authors rely on this architecture along with IPv6 ND to quickly find an alternative parent when required.

Starting from the fact that the RSSI is a poor means to estimate nodes location, KP-RPL [88] is designed as a Kalman positioning based strategy to deal with mobility in RPL network. To alleviate imprecision and the inevitable positioning errors that exist in real-life network deployments, this scheme uses some well-known position nodes (among static ones) assumed to be error free. These nodes also called *anchor*s, are used to estimate the mobile nodes' location according to the Kalman filter algorithm. The latter enhances the positioning accuracy and helps to predict the mobile nodes position by taking their velocity as a parameter. As a result, static nodes build their routes (also referred as *anchor-to-anchor routing*) using the traditional RPL algorithm, whereas mobile nodes (*mobile-to-anchor*) use the KP-RPL. Unfortunately, this model is impracticable as it's very resource intensive and induces a high processing cost, which is not available in an LLN environment.

To sum up, we give below in Table 3 the main RPL versions which improve the protocol for dealing with mobility scenarios.

5. Lessons learned and open issues

Security, mobility and topology optimization are major aspects of LLNs. The great challenge in RPL-based networks is how to keep up-to-date routes and perform a fast rebuild in the case of fast change of topology or varying network conditions. Things are exacerbated if the mobility of nodes is a significant design parameter, in addition to unreliable media and constrained energy of nodes. Many recent works have been proposed by the scientific communities to address the aforementioned topics, but many research challenges remain to be addressed. From the proposals made, many lessons can be learned for future development of the protocol and some guidelines established for new standard documents.

RPL already provides some security mechanisms that enable secure communications between DODAG internal nodes and prevent outside nodes from manipulating or hijacking legitimate traffic. These internal nodes rely on L2 security mechanisms, establish secure channels through pre-installed shared keys or use the existing RPL security modes. Furthermore, there are self-healing measures that reinforce protocol resilience to deal with certain inconsistencies. However, internal nodes are no safe from threats originating from the RPL network itself. So, a malicious node can take advantage of existing "RPL toolkit" (*Trickle* timer, version number, rank) to jeopardize network operations and lifetime. Standard lacks enough measures aiming to protect network from internal threats. Previous works have investigated main threats in this category and propose countermeasures which mitigate them, albeit with an additional overhead. Encryption algorithm and keys management remain real challenges in LLNs due to incurred hardware cost (memory and CPU). Monitoring architectures and IDS have proven to be suitable solutions to counter internal attacks. However, they can induce significant deployment costs (dedicated infrastructure, more powerful nodes) that should be considered. Due to the non-negligible costs of implementing security measures, risk management offers opportunity to dynamically assess incurred risks, then activate and deactivate security mechanisms accordingly. This prevents attacks against network while ensuring a good performance level.

Many emerging IoT applications require real-time data collection and they are expected to integrate various kinds of device including those with mobility capabilities. The RPL default mechanism that governs the dissemination of configuration parameters into network, the *Trickle Algorithm* [31], is more suitable for static environments. Moreover, the mobility of nodes deteriorates the network performance due to continuous changes in the network topology. Efforts outlined above in §4 attempt to address the mobility issues, but much remains to be done. Indeed, RPL in its original version meets the ROLL expectations with regard to deployments where nodes are involved in a *reduced* mobility scenario. However, the protocol experiences many issues (data loss, routing loops, network instabilities, energy expenditure) when the mobility becomes an important design factor. It's therefore necessary to provide suitable mechanisms and propose RPL extensions

Network environment	Enhanced mobility RPL	Ref.	Mobility detection method	Additional mechanisms	Overhead & extra Hardware	Resulting performance
VANET	RPL VANET	[83, 84]	Fixed DIO period	ETX probing	Control msg++	Responsive++ Energy++
	GI-RPL	[85]	DCS strategy, Adaptive DIO period	GIS based	Control msg++ GPS required	Responsive++ Energy++
Mixed nodes environment	*reverse-Trickle* RPL	[27]	DAO mobility flag	*reverse-Trickle* Algo.	Control msg–	Responsive+ Energy– –
	ME-RPL	[25]	# of parent changes	Adaptive DIS interval	Control msg+	Responsive– Energy+
	mod-RPL	[87]	TL adaptive timer	RSSI-based localizat°	Control msg–	Responsive– Energy–
	mRPL	[28]	Link monitoring, Various timers	RSSI Beaconing, *hard* hand-off	Control msg+	Responsive++ Energy+
	mRPL+	[29]	Link monitoring, Various timers	RSSI Beaconing, overhearing, *hard* & *soft* hand-off	Control msg+	Responsive++ Energy+
	Co-RPL	[26]	*corona* mechanism	Path recovery + ND	Control msg++	Responsive– Energy+
	KP-RPL	[88]	*Kalman* filter	Localization through anchors, Blacklisting	No extra msg, Impractical in LLN	Responsive– Energy– –

Table 3: **Summary of proposed RPL extensions that take mobility into account**: For all notations "++" stands for very high, "+" for high, "– –" for very low and "–" for low. Indicated values are considered as extra overhead, energy or responsiveness relative to the legacy RPL deployment.

that deal with this special case. For instance, some network scenarios require all nodes to be in motion, whereas in others only a part of them are mobile. Implementing global rather than local measures (near involved nodes) could lead to an additional overhead affecting the network lifetime. It's then important to delineate the appropriate area where mobility-aware mechanisms will be carried out. In fact, the desired responsiveness level of mobile nodes is achieved at the expense of an additional overhead (in terms of RPL control messages, thus additional energy expenditure). A good trade-off should be found between the desired level of service and the acceptable overhead. Especially, in network environments where the energy of nodes is not a major concern (VANET or rechargeable batteries devices) as is the case in LLN Networks, emphasis should be placed on the nodes responsiveness.

To the best of our knowledge, there is a scarcity of work on RPL targeting the mobility of the sink. Some application scenarios fit well with the latter and deserve a special attention. For instance, a DODAG root embedded in a UAV flying over a harsh area (underground mine, minefield, seismic area). It would activate previously deployed nodes and collect sensed data. Wadhaj *et al.* evaluated RPL in various mobility patterns of the sink node [89]. They found that scenarios using a fixed sink perform much better than mobile ones in all performance aspects (power consumption, PDR and latency). Saad and Tourancheau address the problem in a different way: "where should the sink be placed to improve the network lifetime" [90]. They introduce the *virtual motion* (sink relocation) notion and aim to face the "hot-spot" problem which takes place in nodes near the sink.

Before the official release of the standard, Clausen *et al.* have already highlighted the challenges and problems that RPL should overcome [91]. Although many contributions have been made to improve the protocol, still now some challenges remain open research problems. On the other hand, there is no *parameterization* that fit all LLNs possible use cases. For instance, further investigations should be done to dynamically adapt RPL mobility settings with nodes speed to maintain their responsiveness to an acceptable level compared to the incurred overhead.

Protecting LLN nodes from the outside world relies on data encryption, itself dependent on cryptographic algorithms and key management. Asymmetric cryptography remains an open challenge in LLN and it is currently left out of the standardization process. So, future companion documents of the standard should clearly define how to address this question. As for protecting internal nodes from those already corrupted, dynamic traffic analysis to identify attacks and apply appropriate countermeasures is

used. However, the definition of a threat model applicable to RPL that copes with LLN characteristics should be done. Identity-based (as Sybil attack) as well as attacks using cooperation among intruders (wormhole) are not evaluated for RPL-based networks. This area needs further work. IDS architecture is usually used as a solution to counter several threats in RPL-based networks. Moreover, since nodes are limited in memory size, distributed-IDS should be investigated to overcome related issues and scale.

Link reliability is another concern in LLN, MAC mechanism addresses these issues. However, it would be interesting to exploit cross-layer design when implementing an RPL OF to optimize routing decision. For instance, some authors have shown that it is possible to combine security requirements (trustworthiness) and link reliability to design an OF [24]. Further studies will enable to derive new methods of combination allowing to consider major design factors (security, mobility, application performance) in the routing.

6. Conclusion

This paper reviews recent contributions in this direction, but focusing on those related to the optimization of the topology, security and mobility. An interested reader can also refer to [50] which provides summary statistics on relevant research papers that investigated RPL in recent years and that gives a more general overview on some other topics.

Criteria used to build the routing tree depends on the application goal expressed by way of routing metrics. However, promoting the application objective at the expense of network survivability will lead to the early death of nodes and network holes. Hence, being able to combine application performance and network survivability allow fulfilling QoS in LLN. Additive, lexicographic and Fuzzy-based strategies were proposed as approaches for metrics combination in order to achieve these goals through the RPL OF. Moreover, many new metrics were implemented and evaluated.

We also thoroughly investigated security concerns related to RPL, especially those involving internal nodes as source of the threat. Mitigation strategies provided to counter the identified threats were reviewed and compared. Two trends appear according to how countermeasures are implemented. The are either directly fulfilled on sensor nodes or performed through a dedicated monitoring architecture different from the primary LLN deployed for the application purpose. Both lead to an additional overhead, corresponding to a deployment cost (monitoring architecture) or resources overhead (memory and processing time when using a single network infrastructure) that should be considered when building a secure DODAG topology.

IoT applications, services and users require that RPL capabilities regarding mobility be raised. First works in this direction show that this is done by altering the legacy Trickle timer or by probing some parameters of neighbor nodes. The desired responsiveness and the resulting additional energy cost should be considered prior to deploying an RPL solution consistent with high mobility pattern.

References

[1] W. Kastner, G. Neugschwandtner, S. Soucek, H. M. Newman, Communication systems for building automation and control, Proceedings of the IEEE 93 (6) (2005) 1178–1203. doi:10.1109/JPROC.2005.849726.

[2] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, G. P. Hancke, A survey on smart grid potential applications and communication requirements, Transactions on Industrial Informatics 9 (1) (2013) 28–42. doi:10.1109/TII.2012.2218253.

[3] E. Ancillotti, R. Bruno, M. Conti, The role of the RPL routing protocol for smart grid communications, IEEE Communications Magazine 51 (1) (2013) 75–83. doi:10.1109/MCOM.2013.6400442.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: A survey on enabling technologies, protocols, and applications, IEEE Communications Surveys & Tutorials 17 (4) (2015) 2347–2376. doi:10.1109/COMST.2015.2444095.

[5] E. Borgia, The Internet of Things vision: Key features, applications and open issues, Computer Communications 54 (2014) 1–31. doi:10.1016/j.comcom.2014.09.008.

[6] M. Dohler, T. Watteyne, T. Winter, D. Barthel, Routing Requirements for Urban Low-Power and Lossy Networks, RFC 5548 (Informational) (May 2009). doi:10.17487/RFC5548.

[7] K. Pister, P. Thubert, S. Dwars, T. Phinney, Industrial Routing Requirements in Low-Power and Lossy Networks, RFC 5673 (Informational) (Oct. 2009). doi:10.17487/RFC5673.

[8] A. Brandt, J. Buron, G. Porcu, Home Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5826 (Informational) (Apr. 2010). doi:10.17487/RFC5826.

[9] J. Martocci, P. D. Mil, N. Riou, W. Vermeylen, Building Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5867 (Informational) (Jun. 2010). doi:10.17487/RFC5867.

[10] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944 (Proposed Standard), updated by RFCs 6282, 6775, 8025, 8066 (Sep. 2007). doi:10.17487/RFC4944.

[11] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550 (Proposed Standard) (Mar. 2012). doi:10.17487/RFC6550.

[12] M. Goyal, E. Baccelli, M. Philipp, A. Brandt, J. Martocci, Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks, RFC 6997 (Experimental) (Aug. 2013). doi:10.17487/RFC6997.

[13] M. Goyal, E. Baccelli, A. Brandt, J. Martocci, A Mechanism to Measure the Routing Metrics along a Point-to-Point Route in a Low-Power and Lossy Network, RFC 6998 (Experimental) (Aug. 2013). doi:10.17487/RFC6998.

[14] J. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks, RFC 6551 (Proposed Standard) (Mar. 2012). doi:10.17487/RFC6551.

[15] D. S. De Couto, D. Aguayo, J. C. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proceedings of the 9th ACM Annual International Conference on Mobile Computing and Networking, (MOBICOM), San Diego, CA, USA, 2003, pp. 134–146. doi:10.1145/938985.939000.

[16] P. Thubert, Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL), RFC 6552 (Proposed Standard) (Mar. 2012). doi:10.17487/RFC6552.

[17] O. Gnawali, P. Levis, The Minimum Rank with Hysteresis Objective Function, RFC 6719 (Proposed Standard) (Sep. 2012). doi:10.17487/RFC6719.

[18] P. O. Kamgueu, E. Nataf, T. N. Djotio, O. Festor, Energy-based metric for the routing protocol in low-power and lossy network, in: Proceedings of the 2nd International Conference on Sensor Networks (SENSORNETS), Barcelona, Spain, 2013, pp. 145–148.
URL https://hal.inria.fr/hal-00779519/file/RR-8208.pdf

[19] O. Iova, F. Theoleyre, T. Noël, Using multiparent routing in RPL to increase the stability and the lifetime of the network, Ad Hoc Networks 29 (2015) 45–62. doi:10.1016/j.adhoc.2015.01.020.

[20] P. Di Marco, G. Athanasiou, P. Mekikis, C. Fischione, MAC-aware routing metrics for the Internet of Things, Computer Communications 74 (2016) 77–86. doi:10.1016/j.comcom.2015.05.010.

[21] H. A. Al-Kashoash, Y. Al-Nidawi, A. H. Kemp, Congestion-aware RPL for 6LoWPAN networks, in: Proceedings of the Wireless Telecommunications Symposium (WTS), London, United Kingdom, 2016, pp. 1–6. doi:10.1109/WTS.2016.7482026.

[22] P. O. Kamgueu, E. Nataf, T. N. Djotio, On design and deployment of fuzzy-based metric for routing in low-power and lossy networks, in: Proceedings of the 40th IEEE Local Computer Networks Conference Workshops, LCN, Clearwater Beach, FL, USA, 2015, pp. 789–795. doi:10.1109/LCNW.2015.7365929.

[23] P. Karkazis, P. Trakadas, H. Leligou, L. Sarakis, I. Papaefstathiou, T. B. Zahariadis, Evaluating routing metric composition approaches for QoS differentiation in low power and lossy networks, Wireless Networks 19 (6) (2013) 1269–1284. doi:10.1007/s11276-012-0532-2.

[24] T. N. Velivasaki, P. Karkazis, T. B. Zahariadis, P. Trakadas, C. N. Capsalis, Trust-aware and link-reliable routing metric composition for wireless sensor networks, Trans. Emerging Telecommunications Technologies 25 (5) (2014) 539–554. doi:10.1002/ett.2592.

[25] I. E. Korbi, M. B. Brahim, C. Adjih, L. A. Saidane, Mobility enhanced RPL for wireless sensor networks, in: Proceedings of the Third IEEE International Conference on the Network of the Future (NOF), Gammarth, Tunisia, 2012. doi:10.1109/NOF.2012.6463993.

[26] O. Gaddour, A. Koubâa, M. Abid, Quality-of-service aware routing for static and mobile ipv6-based low-power and lossy sensor networks using RPL, Ad Hoc Networks 33 (2015) 233–256. doi:10.1016/j.adhoc.2015.05.009.

[27] C. Cobarzan, J. Montavont, T. Noël, Analysis and performance evaluation of RPL under mobility, in: IEEE Symposium on Computers and Communications, (ISCC), Funchal, Madeira, Portugal, 2014, pp. 1–6. doi:10.1109/ISCC.2014.6912471.

[28] H. Fotouhi, D. Moreira, M. Alves, mRPL: Boosting mobility in the internet of things, Ad Hoc Networks 26 (2015) 17–35. doi:10.1016/j.adhoc.2014.10.009.

[29] H. Fotouhi, D. Moreira, M. Alves, P. M. Yomsi, mRPL+: A mobility management framework in RPL/6LoWPAN, Computer Communications 104 (2017) 34–54. doi:10.1016/j.comcom.2017.01.020.

[30] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, The Trickle Algorithm, RFC 6206 (Proposed Standard) (Mar. 2011). doi:10.17487/RFC6206.

[31] P. Levis, N. Patel, D. E. Culler, S. Shenker, Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks (awarded best paper!), in: Proceedings of the 1st Symposium on Networked Systems Design and Implementation (NSDI), San Francisco, CA, USA, 2004, pp. 15–28.
URL http://www.usenix.org/events/nsdi04/tech/levisTrickle.html

[32] W. Xie, M. Goyal, S. H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, A. Durresi, A performance analysis of Point-to-Point routing along a directed acyclic graph in low power and lossy networks, in: Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS), Takayama, Gifu, Japan, 2010, pp. 111–116. doi:10.1109/NBiS.2010.65.

[33] E. Baccelli, M. Philipp, M. Goyal, The P2P-RPL routing protocol for IPv6 sensor networks: Testbed experiments, in: Proceedings of the 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 2011, pp. 1–6.
URL https://hal.archives-ouvertes.fr/hal-00651603

[34] J. Ko, J. Jeong, J. Park, J. A. Jun, O. Gnawali, J. Paek, DualMOP-RPL: Supporting multiple modes of downward routing in a single RPL network, ACM Transactions on Sensor Networks (TOSN) 11 (2) (2015) 39:1–39:20. doi:10.1145/2700261.

[35] W. Gan, Z. Shi, C. Zhang, L. Sun, D. Ionescu, MERPL: A more memory-efficient storing mode in RPL, in: Proceedings of the 19th IEEE International Conference on Networks (ICON), Singapore, Singapore, 2013, pp. 1–5. doi:10.1109/ICON.2013.6781985.

[36] C. Király, T. Istomin, O. Iova, G. P. Picco, D-RPL: overcoming memory limitations in RPL point-to-multipoint routing, in: Proceedings of the 40th IEEE Local Computer Networks Conference Workshops, LCN, Clearwater Beach, FL, USA, 2015, pp. 157–160. doi:10.1109/LCN.2015.7366295.
URL https://doi.org/10.1109/LCN.2015.7366295

[37] S. Dawans, S. Duquennoy, O. Bonaventure, On link estimation in dense RPL deployments, in: Proceedings of the 37th Annual IEEE Conference on Local Computer Networks, Workshop Proceedings (LCN), Clearwater Beach, FL, USA, 2012, pp. 952–955. doi:10.1109/LCNW.2012.6424087.

[38] E. Ancillotti, R. Bruno, M. Conti, Reliable data delivery with the IETF routing protocol for low-power and lossy networks, IEEE Trans. Industrial Informatics 10 (3) (2014) 1864–1877. doi:10.1109/TII.2014.2332117.

[39] N. Tsiftes, J. Eriksson, A. Dunkels, Low-power wireless ipv6 routing with contikirpl, in: Proceedings of the 9th International Conference on Information Processing in Sensor Networks (IPSN), Stockholm, Sweden, 2010, pp. 406–407. doi:10.1145/1791212.1791277.
URL http://doi.acm.org/10.1145/1791212.1791277

[40] E. Ancillotti, R. Bruno, M. Conti, E. Mingozzi, C. Vallati, Trickle-L^2: Lightweight link quality estimation through trickle in RPL networks, in: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Sydney, Australia, 2014, pp. 1–9. doi:10.1109/WoWMoM.2014.6918951.

[41] H. Kim, H. Cho, H. Kim, S. Bahk, DT-RPL: diverse bidirectional traffic delivery through RPL routing protocol in low power and lossy networks, Computer Networks 126 (2017) 150–161. doi:10.1016/j.comnet.2017.07.001.

[42] J. Ko, S. Dawson-Haggerty, O. Gnawali, D. Culler, A. Terzis, Evaluating the performance of rpl and 6lowpan in tinyos, in: Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN).

[43] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, A. Terzisand, A. Dunkels, D. Culler, Contikirpl and tinyrpl: Happy together, in: Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN).

[44] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, P. Polakos, Wireless sensor network virtualization: A survey, IEEE Communications Surveys & Tutorials 18 (1) (2016) 553–576. doi:10.1109/COMST.2015.2412971.

[45] H. Kim, H. Cho, M. Lee, J. Paek, J. Ko, S. Bahk, Marketnet: An asymmetric transmission power-based wireless system for managing e-price tags in markets, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys), Seoul, South Korea, 2015, pp. 281–294. doi:10.1145/2809695.2809717.

[46] H. Kim, J. Ko, S. Bahk, Smarter markets for smarter life: Applications, challenges, and deployment experiences, IEEE Communications Magazine 55 (5) (2017) 34–41. doi:10.1109/MCOM.2017.1600260.

[47] E. Ancillotti, C. Vallati, R. Bruno, E. Mingozzi, A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management, Computer Communications 112 (2017) 1–13. doi:10.1016/j.comcom.2017.08.005.

[48] S. Rekik, N. Baccour, M. Jmaiel, K. Drira, Holistic link quality estimation-based routing metric for RPL networks in smart grids, in: Proceedings of the 27th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, Spain, 2016, pp. 1–6. doi:10.1109/PIMRC.2016.7794925.

[49] N. Baccour, A. Koubaa, L. Mottola, M. A. Z. Zamalloa, H. Youssef, C. A. Boano, M. Alves, Radio link quality estimation in wireless sensor networks: A survey, TOSN 8 (4) (2012) 34:1–34:33. doi:10.1145/2240116.2240123.

[50] H. Kim, J. Ko, D. Culler, J. Paek, Challenging the ipv6 routing protocol for low-power and lossy networks (RPL): A survey, IEEE Communications Surveys & Tutorials PP (99) (2017) 1–24. doi:10.1109/COMST.2017.2751617.

[51] S. Moeller, A. Sridharan, B. Krishnamachari, O. Gnawali, Routing without routes: the backpressure collection protocol, in: Proceedings of the 9th International Conference on Information Processing in Sensor Networks (IPSN), Stockholm, Sweden, 2010, pp. 279–290. doi:10.1145/1791212.1791246.

[52] E. Nataf, O. Festor, Accurate online estimation of battery lifetime for wireless sensors network, in: Proceedings of the 2nd International Conference on Sensor Networks (SENSORNETS), INSTICC, ScitePress, Barcelona, Spain, 2013, pp. 59–64.
URL https://doi.org/10.5220/0004312500590064

[53] P. Gonizzi, R. Monica, G. Ferrari, Design and evaluation of a delay-efficient RPL routing metric, in: Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 2013, pp. 1573–1577. doi:10.1109/IWCMC.2013.6583790.

[54] Y. Yang, J. Wang, Design guidelines for routing metrics in multihop wireless networks, in: Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM), Phoenix, AZ, USA, 2008, pp. 1615–1623. doi:10.1109/INFOCOM.2008.222.

[55] J. L. Sobrinho, An algebraic theory of dynamic network routing, IEEE/ACM Trans. Netw. 13 (5) (2005) 1160–1173. doi:10.1109/TNET.2005.857111.

[56] Information processing systems - open systems interconnection – basic reference model - part 2: Security architecture (1989).
URL https://www.iso.org/standard/14256.html

[57] J. Hui, J. Vasseur, The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams, RFC 6553 (Proposed Standard) (Mar. 2012). doi:10.17487/RFC6553.

[58] J. Granjal, E. Monteiro, J. Sá Silva, Security for the Internet of Things: A survey of existing protocols and open research issues, IEEE Communications Surveys and Tutorials 17 (3) (2015) 1294–1312. doi:10.1109/COMST.2015.2388550.

[59] S. Raza, S. Duquennoy, A. Chung, D. Yazar, T. Voigt, U. Roedig, Securing communication in 6LoWPAN with compressed IPsec, in: Proceedings of 7th IEEE International Conference on Distributed Computing in Sensor Systems, (DCOSS), Barcelona, Spain, 2011, pp. 1–8. doi:10.1109/DCOSS.2011.5982177.

[60] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, G. Carle, DTLS based security and two-way authentication for the Internet of Things, Ad Hoc Networks 11 (8) (2013) 2710–2723. doi:10.1016/j.adhoc.2013.05.003.

[61] S. Raza, D. Trabalza, T. Voigt, 6LoWPAN compressed DTLS for CoAP, in: Proceedings of 8th IEEE International Conference on Distributed Computing in Sensor Systems, (DCOSS), Hangzhou, China, 2012, pp. 287–289. doi:10.1109/DCOSS.2012.55.

[62] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, T. Voigt, Secure communication for the Internet of Things - a comparison of link-layer security and IPsec for 6LoWPAN, Security and Communication Networks 7 (12) (2014) 2654–2668. doi:10.1002/sec.406.

[63] P. Pongle, G. Chavan, A survey: Attacks on RPL and 6LoWPAN in IoT, in: Proceeding of IEEE international conference on Pervasive Computing (ICPC), Pune, India, 2015, pp. 1–6. doi:10.1109/PERVASIVE.2015.7087034.

[64] L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the RPL-based Internet of Things, International Journal of Distributed Sensor Networks 9 (2013) 1–11. doi:10.1155/2013/794326.

[65] A. Le, J. Loo, Y. Luo, A. Lasebae, The impacts of internal threats towards routing protocol for low power and lossy network performance, in: IEEE Symposium on Computers and Communications (ISCC), Split, Croatia, 2013, pp. 789–794. doi:10.1109/ISCC.2013.6755045.

[66] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, M. Chai, The impact of rank attack on network topology of routing protocol for low-power and lossy networks, IEEE Sensors Journal 13 (10) (2013) 3685–3692. doi:10.1109/JSEN.2013.2266399.

[67] S. Raza, L. Wallgren, T. Voigt, SVELTE: real-time intrusion detection in the Internet of Things, Ad Hoc Networks 11 (8) (2013) 2661–2674. doi:10.1016/j.adhoc.2013.04.014.

[68] A. Le, J. Loo, A. Lasebae, M. Aiash, Y. Luo, 6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach, International Journal Communication Systems 25 (9) (2012) 1189–1212. doi:10.1002/dac.2356.

[69] A. Le, J. Loo, K. Keong Chai, M. Aiash, A specification-based IDS for detecting attacks on RPL-Based network topology, Information 7 (2) (2016) 1–25. doi:10.3390/info7020025.

[70] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, J. Schönwälder, Mitigation of topological inconsistency attacks in RPL-based low-power lossy networks, Int. Journal of Network Management 25 (5) (2015) 320–339. doi:10.1002/nem.1898.

[71] A. Mayzaud, R. Badonnel, I. Chrisment, A taxonomy of attacks in RPL-based Internet of Things, International Journal of Network Security 18 (3) (2016) 459–473.
URL http://ijns.femto.com.tw/contents/ijns-v18-n3/ijns-2016-v18-n3-p459-473.pdf

[72] K. Chugh, L. Aboubaker, J. Loo, Case study of a black hole attack on 6LoWPAN-RPL, in: The Sixth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), Rome, Italy, 2012, pp. 157–162.

[73] A. Dvir, T. Holczer, L. Buttyán, VeRA - version number and rank authentication in RPL, in: Proceedings of the 8th IEEE International Conference on Mobile Adhoc and Sensor Systems, (MASS), Valencia, Spain, 2011, pp. 709–714. doi:10.1109/MASS.2011.76.

[74] H. Perrey, M. Landsmann, O. Ugus, M. Wählisch, T. C. Schmidt, TRAIL: topology authentication in RPL, in: Proceedings of the International Conference on Embedded Wireless Systems and Networks, (EWSN), Graz, Austria, 2016, pp. 59–64.
URL http://dl.acm.org/citation.cfm?id=2893721

[75] K. Weekly, K. S. J. Pister, Evaluating sinkhole defense techniques in RPL networks, in: Proceedings of the 20th IEEE International Conference on Network Protocols, (ICNP), Austin, TX, USA, 2012, pp. 1–6. doi:10.1109/ICNP.2012.6459948.

[76] K. Zhang, X. Liang, R. Lu, X. Shen, Sybil attacks and their defenses in the internet of things, IEEE Internet of Things Journal 1 (5) (2014) 372–383. doi:10.1109/JIOT.2014.2344013.

[77] F. I. Khan, T. Shon, T. Lee, K. Kim, Merkle tree-based wormhole attack avoidance mechanism in low power and lossy network based networks, Security and Communication Networks 7 (8) (2014) 1292–1309. doi:10.1002/sec.1023.

[78] A. Aris, S. F. Oktug, S. B. O. Yalcin, RPL version number attacks: In-depth study, in: Proceeding of IEEE/IFIP Network Operations and Management Symposium, (NOMS), Istanbul, Turkey, 2016, pp. 776–779. doi:10.1109/NOMS.2016.7502897.

[79] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, J. Schönwälder, A study of RPL DODAG version attacks, in: Proceedings of 8th IFIP International Conference on Autonomous Infrastructure, Management, and Security, (AIMS), Brno, Czech Republic, 2014, pp. 92–104. doi:10.1007/978-3-662-43862-6.

[80] A. Mayzaud, R. Badonnel, I. Chrisment, A distributed monitoring strategy for detecting version number attacks in RPL-based networks, IEEE Trans. Network and Service Management 14 (2) (2017) 472–486. doi:10.1109/TNSM.2017.2705290.

[81] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, M. Richardson, A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs), RFC 7416 (Informational) (Jan. 2015). doi:10.17487/RFC7416.

[82] O. Iova, G. P. Picco, T. Istomin, C. Király, RPL: the routing standard for the internet of things... or is it?, IEEE Communications Magazine 54 (12) (2016) 16–22. doi:10.1109/MCOM.2016.1600397CM.

[83] K. C. Lee, R. Sudhaakar, J. Ning, L. Dai, S. Addepalli, J. P. Vasseur, M. Gerla, A comprehensive evaluation of RPL under mobility, International Journal of Vehicular Technology 2012. doi:10.1155/2012/904308.

[84] K. C. Lee, R. S. Sudhaakar, L. L. Dai, S. Addepalli, M. Gerla, RPL under mobility, in: IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 2012, pp. 300–304. doi:10.1109/CCNC.2012.6181106.

[85] B. Tian, K. M. Hou, H. Shi, X. Liu, Y. Chen, J.-P. Chanet, Application of modified RPL under VANET-WSN communication architecture, in: Proceedings of the Fifth International Conference on Computational and Information Sciences (ICCIS), Shiyang, China, 2013, pp. 1467–1470. doi:10.1109/ICCIS.2013.387.

[86] T. Narten, E. Nordmark, W. Simpson, H. Soliman, Neighbor Discovery for IP version 6 (IPv6), RFC 4861 (Draft Standard), updated by RFCs 5942, 6980, 7048, 7527, 7559, 8028 (Sep. 2007). doi:10.17487/RFC4861.

[87] F. Gara, L. B. Saad, E. B. Hamida, B. Tourancheau, R. B. Ayed, An adaptive timer for RPL to handle mobility in wireless sensor networks, in: Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC), Paphos, Cyprus, 2016, pp. 678–683. doi:10.1109/IWCMC.2016.7577138.

[88] M. Barcelo, A. Correa, J. L. Vicario, A. Morell, X. Vilajosana, Addressing mobility in RPL with position assisted metrics, IEEE Sensors Journal 16 (7) (2016) 2151–2161. doi:10.1109/JSEN.2015.2500916.

[89] I. Wadhaj, I. Kristof, I. Romdhani, A. Y. Al-Dubai, Performance evaluation of the RPL protocol in fixed and mobile sink low-power and lossy-networks, in: Proceedings of the IEEE CIT/IUCC/DASC/PICom, Liverpool, United Kingdom, 2015, pp. 1600–1605. doi:10.1109/CIT/IUCC/DASC/PICOM.2015.241.

[90] L. B. Saad, B. Tourancheau, Sinks mobility strategy in IPv6-based WSNs for network lifetime improvement, in: Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security, (NTM), Paris, France, 2011, pp. 1–5. doi:10.1109/NTMS.2011.5720597.

[91] T. H. Clausen, U. Herberg, M. Philipp, A critical evaluation of the ipv6 routing protocol for low power and lossy networks (RPL), in: IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Shanghai, China, 2011, pp. 365–372. doi:10.1109/WiMOB.2011.6085374.